

클래식 관악악기의 조율기

우주에서 우주

9

```
NotificationCenter.default.addObserver(self,
                                       selector: #selector(setBusInfo),
                                       name: NSNotification.Name(rawValue: "setBusInfo"),
                                       object: nil)

NotificationCenter.default.addObserver(self,
                                       selector: #selector(clickFavoriteButton),
                                       name: NSNotification.Name(rawValue: "favoriteButtonClick"),
                                       object: nil)

NotificationCenter.default.addObserver(self,
                                       selector: #selector(setMainShuttleBus),
                                       name: NSNotification.Name(rawValue: "mainShuttleBusSet"),
                                       object: nil)

NotificationCenter.default.addObserver(self,
                                       selector: #selector(setMainShuttleTime),
                                       name: NSNotification.Name(rawValue: "mainShuttleBusTime"),
                                       object: nil)
```

```
object: nil)
name: NSNotification.Name(rawValue: "mainShuttleBusTime"),
selector: #selector(setMainShuttleTime),
```

```
NotificationCenter.default.addObserver(self,
```

NotificationCenter

클래스간의 의존성을 줄이기 위하여 Controller 객체와 View 객체에 옵저버를 등록 하여 서로 직접적은 의존을 줄일수 있었습니다.

그 결과 코드안에 직접 의존하는 코드가를 많이 줄어 코드 가독성이 좋아지고 클래스의 일을 확실히 할수 있었습니다.

클래스간의 의존성 줄이기

옵저버 패턴 사용

```
NotificationCenter.default.addObserver(self,
                                       selector: #selector(setBusInfo),
                                       name: NSNotification.Name(rawValue: "setBusInfo"),
                                       object: nil)

NotificationCenter.default.addObserver(self,
                                       selector: #selector(clickFavoriteButton),
                                       name: NSNotification.Name(rawValue: "favoriteButtonClick"),
                                       object: nil)

NotificationCenter.default.addObserver(self,
                                       selector: #selector(setMainShuttleBus),
                                       name: NSNotification.Name(rawValue: "mainShuttleBusSet"),
                                       object: nil)

NotificationCenter.default.addObserver(self,
                                       selector: #selector(setMainShuttleTime),
                                       name: NSNotification.Name(rawValue: "mainShuttleBusTime"),
                                       object: nil)
```

```
object: nil)
name: NSNotification.Name(rawValue: "mainShuttleBusTime"),
selector: #selector(setMainShuttleTime)
NotificationCenter.default.addObserver(self,
```

NotificationCenter

클래스간의 의존성을 줄이기 위하여 Controller 객체와 View 객체에 옵저버를 등록 하여 서로 직접적인 의존을 줄일수 있었습니다.

그 결과 코드안에 직접 의존하는 코드가 많이 줄어 코드 가독성이 좋아지고 클래스의 일을 확실히 할수 있었습니다.

서버 통신

Alamofire, Swifty 오픈소스 활용



ServerRepository 클래스를 만들 서버통신 부분을 따로 관리 했습니다.
Escaping Closure를 이용하여 비동기 통신이 가능하도록 구현하였습니다.
JSON 형태로 통신을 Swifty오픈소스를 이용하여 보다 쉽게 맵핑하였습니다.